REMARKS

Applicant respectfully requests reconsideration of this application in view of the following remarks. In general, Applicant traverses all of the assertions in the Office Action. For the Examiner's convenience and reference, Applicant's remarks are presented in substantially the same order in which the corresponding issues were raised in the Office Action.

As a preliminary matter, Applicant appreciates the Examiner's assistance in recognizing that an incomplete copy of foreign patent document WO 01/75620 A1 was submitted. Applicant resubmits a copy of this document, in its entirety, with this response and respectfully requests that the Examiner consider the complete document at this time. Although Applicant believes no additional fee is required for this submission, please charge any required fees related to this submission to the deposit account identified at the conclusion of this response.

Status of the Claims

Claims 1-3 and 5-21 are pending. Claims 1, 6-8, and 21 are currently amended. Claims 12, 15, and 16 are canceled. Claims 22-29 are added. No new matter has been added.

Summary of the Office Action

The Office Action states that claims 1-3, 5-14, and 17-21 stand rejected under 35 U.S.C. § 102(e) as being anticipated by or, in the alternative, under 35 U.S.C. 103(a) as obvious over U.S. Patent No. 6,510,497 to Strongin et al. (hereinafter "Strongin") in view of PCT Publication No. 07/75620 to Novak (hereinafter "Novak"). From this, it appears that the Office Action makes the following separate rejections:

Claims 1-3, 5-14, and 17-21 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Strongin.

Claims 1-3, 5, 8-11, 13-16, and 21 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Novak.

Claims 1-3, 5-14, and 17-21 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Strongin in view of Novak

Claims 15-16 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Strongin.

Response to Rejections

Claims 1-3 and 5-21 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Strongin and/or Novak. Claims 1-3 and 5-21 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Strongin alone, or Strongin in view of Novak. Applicant respectfully requests withdrawal of these rejections because the cited references fail to disclose all of the limitations of the claims.

Strongin is directed to a memory controller to schedule accesses to system memory. Strongin, Abstract. Strongin teaches that the memory controller, via the memory arbiter and memory state tracking unit, receives information about the status of the system memory. Strongin, col. 11, lines 43-52. The memory arbiter then uses this memory state information to schedule pending memory operations according to a scheduling hierarchy. Strongin, col. 12, lines 3-12. The scheduling hierarchy includes six hierarchical levels that depend on one or more memory operation characteristics. Strongin, col. 12, lines 11-44. However, Strongin fails to disclose tracking a count of a number of scheduled requests. Nevertheless, even if Strongin were to disclose tracking a count of a number of scheduled requests, as suggested by the Examiner, Strongin fails to disclose a configurable switch point.

Strongin does not disclose a switch point because Strongin merely describes using a hierarchy to schedule pending memory operations. Although the Office Action states that Strongin discloses a switch point at column 12, lines 20-35, this reference merely describes the second, third, and fourth, tiers of the hierarchy. Memory requests within each of these tiers are scheduled in order, but these tiers do not describe a switch point. Rather, this reference merely describes the types of memory operations which are categorized in each tier. In particular, the second tier includes pending non-speculative memory operations in the opposite bus direction; the third tier includes non-speculative operations in the same bus direction for pages not currently open; and the fourth tier includes non-speculative operations in the opposite bus direction for pages not currently open. Strongin, col. 12, lines 21-36. The Office Action also asserts that Strongin

discloses a switch point at column 18, lines 22-35. Again, this reference merely describes using a hierarchy to schedule the pending memory operations, but does not describe a switch point. In particular, the cited reference merely discloses scheduling first tier operation (open pages in the same bus direction) before scheduling operations from the second, third, and fourth tiers (closed pages or a different bus direction). Strongin, col. 18, lines 29-35; see also col. 12, lines 11-36.

The Office Action further states that "the 'switch point' is when the pending requests consistent with the memory bus direction are issued, and the bus direction reverses, or switches . . . ." However, this statement places the term "switch point" out of context because it attempts to interpret "switch point" as merely a point in time. If the switch point were merely a point in time, then the description in paragraph [0031] and other language of the specification would not make sense because a count of memory requests is compared to the switch point. Moreover, the language of claim 1 would not make sense because "the switch point includes a threshold value."

Additionally, even if Strongin were to disclose a switch point, Strongin does not disclose a configurable switch point because there is no disclosure of how "when the pending requests consistent with the memory bus direction are issued, and the bus direction reverses, or switches" might be configured. In fact, the method of Strongin for scheduling memory operations is fixed—not configurable—according to the scheduling hierarchy. Strongin explicitly states "the pending memory operations are scheduled for execution in the following hierarchy" and proceeds to explain the six tiers. Strongin, col. 12, lines 11-44. Strongin does not disclose any way to change or reconfigure the tiers of the schedule hierarchy. Thus, Strongin does not disclose a configurable switch point.

Novak also fails to teach a configurable switch point. Although Novak teaches counting requests to prevent starvation of service to requests with lesser priority, Novak does not describe configuring or allowing the PH request limit to be configured. Rather, Novak merely teaches that the PH request limit is set at 32 so that no more than 32 PH requests are serviced consecutively. Thus, Novak does not disclose a configurable switch point.

In contrast, claim 1 includes "storing a configurable switch point, wherein the configurable switch point includes a threshold value to indicate when to switch from the

current state to an alternate state of the device." For the reasons stated above, Strongin and Novak fail to disclose all of the limitations of claim 1. In particular, Strongin and Novak do not disclose storing a configurable switch point, wherein the configurable switch point includes a threshold value to indicate when to switch from the current state to an alternate state of the device. Given that the cited references fail to disclose all of the limitations of the claim, Applicant respectfully submits that claim 1 is patentable over the cited references. Accordingly, Applicant requests that the rejections of claim 1 under 35 U.S.C. §§ 102(e) and 103(a) be withdrawn.

Each of independent claims 6, 8, and 21 includes a similar limitation to the limitation of claim 1. Given that the cited references fail to disclose at least the described limitations, Applicant respectfully submits that independent claims 6, 8, and 21 are each patentable over the cited references. Furthermore, independent claims 6, 8, and 21 may be patentable over the cited references for additional reasons. Accordingly, Applicant requests that the rejections of claims 6, 8, and 21 under 35 U.S.C. §§ 102(e) and 103(a) be withdrawn.

Given that claims 2-3 and 5 depend from independent claim 1, which is patentable over the cited reference, Applicant respectfully submits that dependent claims 2-3 and 5 are also patentable over the cited reference. Accordingly, Applicant requests that the rejection of claims 2-3 and 5 under 35 U.S.C. §§ 102(e) and 103(a) be withdrawn.

Given that claims 7, 18, 20, and 22-25 depend from independent claim 6, which is patentable over the cited reference, Applicant respectfully submits that dependent claims 7, 18, 20, and 22-25 are also patentable over the cited reference. Accordingly, Applicant requests that the rejection of claims 7, 18, and 20 under 35 U.S.C. §§ 102(e) and 103(a) be withdrawn.

Given that claims 9-11, 13-14, 17, 19, and 26-29 depend from independent claim 8, which is patentable over the cited reference, Applicant respectfully submits that dependent claims 9-11, 13-14, 17, 19, and 26-29 are also patentable over the cited reference. Accordingly, Applicant requests that the rejection of claims 9-11, 13-14, 17, and 19 under 35 U.S.C. §§ 102(e) and 103(a) be withdrawn.

Response to the Official Notice

The Office Action appears to assert that the subject matter of claims 15 and 16 has been taken as admitted prior art because Applicant did not traverse the Official Notice taken in the previous Office Action. Applicant respectfully requests that the Examiner clarify this position because the previous Office Action did not take Official Notice of the subject matter of claims 15 and 16. Even though claims 15 and 16 have been canceled, Applicant respectfully requests that the Examiner acknowledge that the material for which Official Notice was taken is different from the subject matter of claims 15 and 16.

The Office Action mailed August 11, 2005, stated that the Examiner took Official Notice that "[i]t is extremely well known to those of even rudimentary skill in the art that a register may be employed to count requests in a buffer or queue of request to inform the system how many particular requests may be present." While a register may be employed to count requests in a buffer or queue, the subject matter of claims 15 and 16 was not directed to merely using a register to store a count request. Rather, claims 15 and 16 each recited "a register to store the threshold number of counts to establish the switch point." Applicant respectfully submits that merely storing a count, or quantity, of actual requests in queue is not the same as storing a threshold number of counts. Additionally, merely storing a count or requests in a queue is insufficient to establish a switch point, as recited in the claims.

Given that the subject matter of claims 15 and 16 is different from merely employing a register to count requests in a buffer or queue, the Official Notice does not render the subject matter of claims 15 and 16 as admitted prior art. At most, the Official Notice addresses a register to count requests in a buffer or queue, which is different from "a register to store the threshold number of counts to establish the switch point."

CONCLUSION

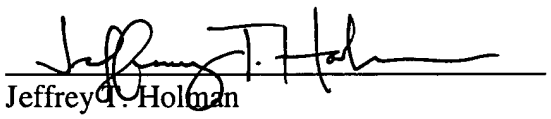
It is respectfully submitted that in view of the amendments and remarks set forth herein, the rejections have been overcome. If the Examiner believes a telephone interview would expedite the prosecution of this application, the Examiner is invited to contact Jeffrey Holman at (408) 720-8300.

If there are any additional charges, please charge them to Deposit Account No. 02-2666.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Date: 6/5/06



Jeffrey T. Holman
Reg. No. 51,812

12400 Wilshire Blvd.
Seventh Floor
Los Angeles, CA 90025-1026
(408) 720-8300

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 October 2001 (11.10.2001)

PCT

(10) International Publication Number
WO 01/75620 A1

(51) International Patent Classification[7]: G06F 13/18, 13/40

(21) International Application Number: PCT/US00/31963

(22) International Filing Date:
21 November 2000 (21.11.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/541,234 3 April 2000 (03.04.2000) US

(71) Applicant: ADVANCED MICRO DEVICES, INC. [US/US]; One AMD Place, Mail Stop 68, Sunnyvale, CA 94088-3453 (US).

(72) Inventor: NOVAK, Steve; AM Haideberg 13, 01468 Boxdorf (DE).

(74) Agent: APPERLEY, Elizabeth, A.; Advanced Micro Devices, Inc., 5204 East Ben White Boulevard M/S 562, Austin, TX 78741 (US).

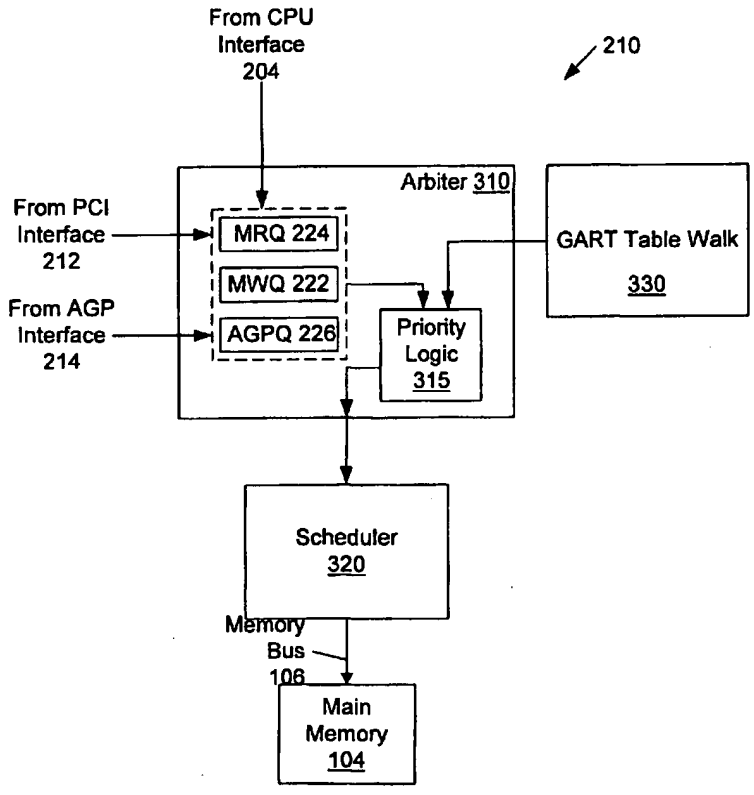
(81) Designated States *(national)*: CN, JP, KR, SG.

(84) Designated States *(regional)*: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: BUS BRIDGE INCLUDING A MEMORY CONTROLLER HAVING AN IMPROVED MEMORY REQUEST ARBITRATION MECHANISM



(57) Abstract: A bus bridge (102) including a memory controller (210) having an improved memory request arbitration mechanism is disclosed. The memory controller (210) receives various requests to read from or write to the main memory (104). In a particular embodiment, the memory controller (210) may be configured to categorize these incoming requests into a page hit request, a page miss bank request, a page miss-different chip select request and a page conflict request. The memory controller (210) may be configured to prioritize these requests based on latency. Page hit requests have a higher arbitration priority than page miss bank requests which have a higher arbitration priority than page miss different chip-select requests which have a higher arbitration priority than page conflict requests. Since the memory controller (210) services requests based on priority, it enhances the utilization of a memory bus (106), such as an SDRAM bus.

WO 01/75620 A1

TITLE: BUS BRIDGE INCLUDING A MEMORY CONTROLLER HAVING AN IMPROVED MEMORY REQUEST ARBITRATION MECHANISM

BACKGROUND OF THE INVENTION

1. Technical Field

This invention relates to computer systems and, more particularly, to arbitration mechanisms employed within memory controllers.

2. Background Art

Computer systems typically include a plurality of devices that are interconnected by one or more buses. For example, many conventional computer systems comprise a processor that is coupled to a main memory through a bus bridge. The bus bridge may be coupled to various peripheral devices, such as network interface cards, video accelerators, audio cards, SCSI adapters, telephony cards, etc. through a high bandwidth local expansion bus, such as the Peripheral Component Interconnect (PCI) bus. The bus bridge may further be coupled to a graphics controller through a second expansion bus, such as an Advanced Graphics Port (AGP) bus.

The bus bridge may further include a memory controller. A memory controller receives various requests to read from or write to the main memory. The memory controller may receive these incoming requests from various sources, such as the processor or a peripheral connected through one of the expansion buses. Internally, the bus bridge may include a processor bus interface, a PCI bus interface and an AGP bus interface to provide appropriate interfaces to the memory controller.

In some implementations, the main memory may be implemented using synchronous dynamic random access memory (SDRAM). Figure 1 illustrates an exemplary memory subsystem 100 including a memory controller 10 coupled to a plurality of SDRAM modules 16A-C through SDRAM bus 12. The plurality of SDRAM modules 16A-C are connected to SDRAM bus 12 through memory expansion slots designated as 14A-C. Each SDRAM module illustratively includes a plurality of banks (i.e., denoted Bank 1- Bank 4). Each bank includes a plurality of pages. Memory controller 10 receives a memory access address that dictates which page will be accessed in a given bank within a given module. The access address determines which chip select signal, CS1-3, will be asserted in order to select the particular module that contains the page and bank to be accessed.

Prior to reading from or writing to a particular location in the SDRAM, the page in which the particular location resides must be activated or opened. A page is opened by an activation cycle initiated by memory controller 10 on SDRAM bus 12. After the page is opened, the read or write is performed by a read/write cycle initiated by memory controller 10. Since two pages in a given bank cannot be opened simultaneously, prior to opening a given page, another page in the bank may need to be closed. The memory controller 10 closes a page by initiating a precharge cycle on SDRAM bus 12.

The latency for servicing a particular read or write request depends upon the state of memory subsystem 100 when a corresponding SDRAM cycle(s) (e.g. read/write cycle, activation cycle, and precharge cycle) is driven by memory controller 10 on SDRAM bus 12. For example, consider a situation in which memory controller 10 receives a request to read from or write to an active page in a particular bank.

Since the page is already active, memory controller 10 initiates only a read/write cycle to read from or write to the selected page. A request to a page that is already active is commonly referred to as a page hit request. Page hit requests have the lowest latency since only one cycle is initiated by memory controller 10.

Alternatively, memory controller 10 may receive a request to open a page in a bank that has no activated page. The selected page is opened or activated by an activation cycle initiated by memory controller 10. A read/write cycle is then initiated by memory controller 10 to read from or write to the open page. A request to open a page in a bank that has no activated page is commonly referred to as a page miss bank request. Page miss bank requests have the second lowest latency because two separate cycles are initiated by memory controller 10.

Alternatively, memory controller 10 may receive a request to open a page in a different module. In what is commonly referred to as a one cycle turnaround bubble, the currently active module is first deselected and then the module that contains the page to be accessed is selected. A one cycle turnaround bubble is necessitated to ensure that the two modules, i.e. the deselected and selected modules, are not driving SDRAM bus 106 simultaneously. After the proper module is selected, the selected page is opened by an activation cycle. A read/write cycle is then initiated to read from or write to the open page. A request to open a page in a different module is commonly referred to as a page miss-different chip select request. Page miss-different chip select requests have the third lowest latency because of the one cycle turnaround bubble and the two separate cycles that are initiated by memory controller 10.

Alternatively, memory controller 10 may receive a request to open a page not active in an already active bank. The currently active page is closed or deactivated by a precharge cycle initiated by memory controller 10. The selected page is then opened or activated by an activation cycle. A separate cycle, the read/write cycle, is then initiated to read from or write to the open page. A request to open a page not active in an already active bank is commonly referred to as a page conflict request. Page conflict requests have the longest latency because three separate cycles are initiated by memory controller 10.

In some applications, memory controller 10 may be configured such that incoming requests are serviced in order of receipt. A disadvantage of memory controller 10 servicing requests in order of receipt is that a request that may have been categorized as one particular type of request may become categorized as a different type of request that has a longer latency. For example, if memory controller 10 receives a page conflict request followed by three consecutive page hit requests and services them in order of receipt, then memory controller 10 first services the page conflict request. However, after servicing the page conflict request, the page hit requests may become page conflict requests. Memory controller 10 services a page conflict request by initiating a precharge cycle to close the current page and initiating an activation cycle to activate the selected page followed by initiating a read/write cycle. However, the selected page that is opened is a different page than the page requested by the page hit requests. The page hit requests may subsequently become page conflict requests. That is, the former page hit requests may request to read from or write to a formerly active page that is no longer active. The current page may then have to be closed and a new page activated increasing the time to service the requests. A memory controller 10 servicing incoming requests in order of receipt may not optimally utilize the memory bus. Therefore it is desirable to access the SDRAM more efficiently.

DISCLOSURE OF INVENTION

The problems outlined above may in large part be solved by a bus bridge including a memory controller having an improved memory request arbitration mechanism. In one embodiment, a bus bridge comprises a memory controller coupled to a main memory through a memory bus. The bus bridge may further comprise a processor bus interface and a first peripheral bus interface both coupled to the memory controller. The processor bus interface provides an interface between the memory controller and a processor. The first peripheral bus interface provides an interface between the memory controller and the first peripheral bus.

The memory controller receives various requests to read from or write to the main memory. The memory controller may receive these incoming requests from the processor bus interface and the first peripheral bus interface. In a particular embodiment, the memory controller is configured to accept and arbitrate among these incoming requests. The memory controller may be configured to categorize these incoming requests into a page hit request, a page miss bank request and a page miss-different chip select request. A page hit request has the lowest latency because the page requested is already active. However, the page miss bank request has a longer latency than the page hit request because an activation cycle is required to open the selected page followed by a read/write cycle to read from or write to the selected page. The page miss-different chip select request has a longer latency than the page hit and page miss bank request because there is a one cycle turnaround bubble followed by an activation and read/write cycle. The memory controller may be configured to prioritize these requests based on latency. Subsequently, the page hit request has a higher arbitration priority than page miss bank and page miss different chip-select request. Page miss bank request has a higher arbitration priority than page miss different chip-select request. The memory controller will then service those requests with a higher arbitration priority than those requests with a lower arbitration priority and hence may improve the utilization of the memory bus, such as an SDRAM bus.

In another particular embodiment, the memory controller may accept and arbitrate among an additional incoming request such as a page conflict request. A page conflict request has a longer latency than the page hit, page miss bank and page miss-different chip select request because the memory controller initiates a precharge, activation and read/write cycle. Consequently, the memory controller may be further configured to provide the page miss-different chip select request a higher arbitration priority than the page conflict request. The memory controller will then service those requests with a higher arbitration priority than those requests with a lower arbitration priority and hence may improve the utilization of the memory bus, such as an SDRAM bus.

In another particular embodiment, the memory controller may be configured to prevent the servicing of page hit requests repeatedly at the exclusion of servicing requests with a lesser priority. The memory controller includes a programmable logic unit which limits the number of page hit requests serviced consecutively from one to thirty-two. The programmable logic unit includes a register configured to store the page hit limit and a counter coupled to the register. The counter is configured to count the number of priority hits that were serviced by the memory controller consecutively.

BRIEF DESCRIPTION OF DRAWINGS

Other objects and advantages of the invention will become apparent upon reading the following detailed description and upon reference to the accompanying drawings in which:

Figure 1 illustrates an exemplary memory subsystem.

Figure 2 is a block diagram of one embodiment of a computer system including a bus bridge.

Figure 3 is a block diagram of one embodiment of a bus bridge.

Figure 4 is a block diagram of one embodiment of a memory controller.

Figure 5 is a block diagram of one embodiment of a priority logic block of the arbiter of Figure 4.

Figure 6 is a block diagram of one embodiment of the categorizing logic of the categorizing multiplexer of Figure 5.

Figure 7 illustrates one embodiment of the prioritizing logic of the priority select multiplexer of Figure 5.

Figure 8 illustrates one embodiment of a scheduler in a memory controller.

While the invention is susceptible to various modifications and alternative forms, specific embodiments thereof are shown by way of example in the drawings and will herein be described in detail. It should be understood, however, that the drawings and detailed description thereto are not intended to limit the invention to the particular form disclosed, but on the contrary, the invention is to cover all modifications, equivalents and alternatives falling within the spirit and scope of the present invention as defined by the appended claims.

MODE(S) FOR CARRYING OUT THE INVENTION

Figure 2 illustrates one embodiment of a computer system. Computer system 200 includes a processor 101 coupled to a variety of system components through a bus bridge 102. In computer system 200, a main memory 104 is coupled to bus bridge 102 through a memory bus 106. A graphics controller 108 is coupled to bus bridge 102 through an AGP bus 110. Furthermore, a plurality of PCI (Peripheral Component Interconnect) devices 112A-112B are coupled to bus bridge 102 through a PCI bus 114.

Processor 101 is illustrative of, for example, an Athlon™ Microprocessor. It is understood, however, that in other embodiments of computer system 200, alternative types of microprocessors may be employed. In other embodiments, an external cache unit (not shown) may further be coupled to processor 101.

Bus bridge 102 provides an interface between processor 101, main memory 104, graphics controller 108, and devices attached to PCI bus 114. When an operation is received from one of the devices connected to bus bridge 102, bus bridge 102 routes the operation to the targeted device. Bus bridge 102 generally translates an operation from the protocol used by the source device or bus to the protocol used by the target device or bus.

Main memory 104 is a memory in which application programs and data are stored. Processor 101 executes primarily out of main memory 104. In one embodiment, main memory 104 may be dynamic random access memory (DRAM) or preferably, in other embodiments, main memory 104 may be synchronous dynamic random access memory (SDRAM).

PCI devices 112A-112B are illustrative of a variety of peripheral devices such as, for example, network interface cards, video accelerators, audio cards, hard or floppy disk drives, small computer systems interface (SCSI) adapters and telephony cards. Similarly, industry standard architecture (ISA) device 118 is illustrative of one of a variety of peripheral devices such as a modem, a sound card, and a variety of data acquisition cards such as general purpose interface bus (GPIB) or field bus interface cards.

Graphics controller 108 is provided to control the rendering and images for a display device. Graphics controller 108 may embody a typical graphics accelerator generally known in the art to render three-dimensional

data structures which can be effectively shifted into and out of main memory 104. Graphics controller 108 may therefore be a master of AGP bus 110 in that it can request and receive access to a target interface within bus bridge 102 and thereby obtain access to main memory 104.

It is noted that while AGP bus 110 and PCI bus 114 have been used as examples in the above description, any peripheral bus standard may be used. As will be described in more detail below, bus bridge 102 may receive requests from CPU bus 103, PCI bus 114 and AGP bus 110 to read from or write to main memory 104. Bus bridge 102 may categorize these requests according to a prioritization scheme which takes into account the priority of each request relative to each other request and the state of main memory 104 at the time of the request. In this manner, bus bridge 102 may advantageously optimize the utilization of the bandwidth of memory bus 106, thus increasing overall performance of the computer system.

Referring to Figure 3, one embodiment of a bus bridge is illustrated. Circuit components that correspond to those shown in FIG. 2 are numbered identically for simplicity and clarity. Bus bridge 102 includes a CPU interface 204, a PCI interface 212, an AGP interface 214 and a memory controller 210. Memory controller 210 is coupled to CPU interface 204, PCI interface 212 and AGP interface 214. Memory controller 210 is also coupled to a main memory 104 through memory bus 106. It is noted that the components of bus bridge 102 may be embodied on a single integrated circuit chip or they may be embodied within other components of computer system 200 of Figure 2.

As described further below, memory controller 210 uses a plurality of queues as a request buffering mechanism. The queues are configured to receive requests from CPU bus 103, PCI bus 114 and AGP bus 110 through CPU interface 204, PCI interface 212 and AGP interface 214, respectively.

Referring now to Figure 4, a block diagram of one embodiment of a memory controller is shown. Circuit components that correspond to those shown in Figure 2 and Figure 3 are numbered identically for simplicity and clarity. Memory controller 210 comprises an arbiter 310 which is coupled to a scheduler 320 and a graphics address remapping table (GART) table walk 330. Arbiter 310 is also coupled to CPU interface 204, PCI interface 212 and AGP interface 214. Scheduler 320 is coupled to main memory 104 through memory bus 106.

Arbiter 310 includes a memory read queue (MRQ) 224, a memory write queue (MWQ) 222, an advanced graphics port queue (AGPQ) 226 and a priority logic block 315. Arbiter 310 may be configured to receive external memory requests from CPU interface 204, PCI interface 212 and AGP interface 214. Arbiter 310 may also be configured to receive one internal memory request from GART table walk 330 and one bypass request (BYP). The external requests are stored in one of MRQ 224, MWQ 224 or AGPQ 226. MRQ 224 stores requests to read from main memory 104. MWQ 222 stores requests to write to main memory 104 and AGPQ 226 stores requests to read from or write to main memory 104. It is noted that in one particular embodiment, MRQ 224 may be embodied in four separate buffers to accommodate up to four memory read requests. It is described here as one queue for the sake of simplicity. The size of MWQ is one entry and the size of AGPQ is one entry. It is noted however that in various other embodiments, the sizes of MRQ 224, MWQ 222 and AGPQ 226 may be different. Once the requests are in their respective queues, they are then categorized further depending on the state of main memory 104. Priority logic block 315 selects and presents one of the requests to scheduler 320 according to a fixed priority which is described further below. Scheduler 320 then schedules the request, either read or write, onto memory bus 106.

In this particular embodiment, main memory 104 is embodied in SDRAM. As described above, SDRAM accesses typically occur in page mode. These page mode accesses may be categorized into page hit requests (PH), page miss bank requests (PMb), page miss different chip select requests (PMc) and page conflict requests (PC). The priority given to the requests is based on the latency associated with each type of request.

Turning now to Figure 5, a block diagram of one embodiment of a priority logic block of arbiter 310 of Figure 4 is shown. Circuit components that correspond to those shown in Figure 4 are numbered identically for simplicity and clarity. Priority logic block 315 includes categorizing multiplexers 410A, 410B, 410C and 410D which use combinational logic to categorize a plurality of requests received from MRQ 224, MWQ 222 and AGPQ 226 of Figure 4 into the four categories described above: page hit (PH), page miss bank (PMB), page miss-different chip select (PMC) and page conflict (PC). For this discussion, categorizing multiplexers 410A, 410B, 410C and 410D may each be referred to as categorizing multiplexer 410. It is noted that in other embodiments, there may be additional or fewer number of categorizing multiplexers 410 depending on the number of categories. Categorizing multiplexers 410 are configured to select one request from each category, (e.g. PH, PMb, PMc and PC), based on a categorizing scheme as illustrated in Figure 6. The outputs of categorizing multiplexers 410 are input to a priority select multiplexer 420.

Priority select multiplexer 420 selects the next request to be scheduled to scheduler 320. Priority select multiplexer 420 selects from the following external/internal requests: an opportunistic precharge (OP), a high priority request from AGP interface 214 (Hi AGP), a high priority memory write request(Hi MWQ), an internal request from Gart Table Walk (GTW) 330 as well as the four categorized requests (PH, PMb, PMc, PC) from categorizing multiplexers 410, and an internal idle precharge signal. Once the requests are prioritized, priority select multiplexer 420 is configured to pass the one highest priority request to scheduler 320. The priority selection method described below in the description of Figure 7.

Priority select multiplexer 420 prioritizes the output requests from categorizing multiplexers 410 (e.g. PH's, PMb's, PMc's and PC's) based on the latency exhibited by these requests. As stated earlier, PH requests receive a higher arbitration priority over PMb, PMc and PC requests. However, to prevent starvation of service to requests with a lesser priority, priority select multiplexer 420 may be configured to stop servicing PH requests when a predetermined limit is reached. Priority select multiplexer 420 may include a programmable logic unit 421 which limits the number of page hit requests serviced consecutively from one to thirty-two. Programmable logic unit 421 includes a limit register 422 configured to store the PH request limit and a counter 423 coupled to limit register 422. Counter 423 is configured to count the number of priority hits that were serviced by scheduler 320 consecutively. Once counter 423 reaches the programmable limit, priority select multiplexer 420 may select a different request, for example, a PMb or a PMc. Counter 423 will then be reset to zero and will begin recounting the number of priority hits serviced by scheduler 320 consecutively.

Referring to Figure 6, a block diagram of one embodiment of the categorizing logic of the categorizing multiplexer of Figure 5 is shown. Circuit components that correspond to those shown in Figure 5 are numbered identically for simplicity and clarity. Categorizing multiplexer 410 uses combinational logic comprising a plurality of multiplexers to select the highest priority request. It is noted that other embodiments may use other logic blocks and configurations.

Categorizing multiplexer 410 also includes an internal signal to indicate that neither AGPQ request nor MWQ request is valid. This internal request is called an invalid category pointer (INV). Requests from memory read queue 0 (MRQ0), are given the highest priority; whereas, INV is given the lowest priority. Multiplexer 610 provides requests from MRQ0 a higher priority than requests from memory read queue 1, MRQ1. Multiplexer 620 provides requests from memory read queue 2, MRQ2, a higher priority than requests from memory read queue 3, MRQ3. Multiplexer 630 provides requests from MRQ0 and MRQ1 a higher priority than requests from MRQ2 and MRQ3. Multiplexer 640 provides requests from MWQ, a higher priority than requests from AGPQ. Multiplexer 640 further provides requests from AGPQ a higher priority than invalid category pointer. Multiplexer 650 provides requests from MRQ0-3, a higher priority than requests from MWQ and AGPQ. The output of categorizing multiplexer 410 is a valid category request pointer for input into priority select multiplexer 420 of Figure 4.

Turning now to Figure 7, one embodiment of the prioritizing logic of priority select multiplexer 420 of Figure 5 is illustrated. The combinational logic of priority select multiplexer 420 comprises a plurality of two input multiplexers. It is noted that in other embodiments, other logic configurations may be used. Priority select multiplexer 420 assigns its incoming requests a priority from highest to lowest as follows: bypass (BYP), OP, Hi MWQ, Hi AGP, GTW, PH, PMb, PMc, PC and idle precharge.

Multiplexer 710 gives OP a higher priority than Hi MWQ. Multiplexer 720 gives Hi AGP a higher priority than GTW. Multiplexer 730 gives PC a higher priority than idle precharge. Multiplexer 740 gives PH a higher priority over PMb. Multiplexer 750 gives PMc a higher priority than both PC and Idle Precharge. Multiplexer 760 gives Precharge and Hi MWQ a higher priority than Hi AGP and GTW. Multiplexer 770 gives PH and PMb a higher priority than PMc, PC and Idle Precharge. Multiplexer 780 gives Precharge, Hi MWQ, Hi AGP and GTW a higher priority than PH, PMB, PMC, PC and idle precharge. Multiplexer 790 gives Bypass the highest priority. The output of multiplexer 790 is the selected request sent to scheduler 320.

As described in more detail below, the BYP request is issued only when there are no current valid chip selects. This may occur when memory controller 210 is completely idle. Since no chip select signal is currently valid, a PMc request will be treated as a PMb request. Whenever a BYP request is accepted, the BYP request may pass directly through scheduler 320 of Figure 5 and onto the memory bus 106 of Figure 4 in one cycle.

BYP is given highest priority since the request may be placed on memory bus 106 in one cycle. However, the instances in which memory controller is completely idle may be rare. The OP request is given second highest priority of all other requests. OP is issued only one cycle after an 8 quad word PH or PMb request has been accepted to reduce the probability that a PC request will occur. The idle precharge request is issued when no new valid requests have been presented to arbiter 310 for more than a programmable number of cycles and therefore it is given the lowest priority of all requests.

Figure 8 illustrates one embodiment of scheduler 320 of Figure 5. Circuit components that correspond to those shown in Figures 2 through 7 are numbered identically for simplicity and clarity. Scheduler 320 schedules the requests received from priority select multiplexer 420 of Figure 5 onto memory bus 106 of Figure 8. Scheduler 320 comprises three queues coupled to a priority multiplexer 840. The three queues are: a pre-charge queue (PQ) 820, an activate queue (AQ) 810 and a read/write queue (RWQ) 830. Scheduler 320 may be configured to provide PQ 820 with the highest priority followed by AQ 810 followed by RWQ 830.

A request to initiate an OP cycle is loaded into PQ 820. A request to initiate an activation cycle is loaded into AQ 810. A request to initiate a read/write cycle is loaded into RWQ 830. It is important to note that in one particular embodiment, the size of AQ 810 and PQ 820 is one deep. The size of RWQ 830 is 3 deep. The depth of one forces scheduler to accept the next request at the latest possible time before the last request is finished processing and memory bus 106 becomes idle. Waiting until the latest possible time to accept the next request from arbiter 310 of Figure 5 gives arbiter 310 the opportunity to examine more requests, thereby increasing the probability that a PH or PMb request will be presented. It is noted however, that the size of the queues in scheduler 320 of Figure 8 may be varied in different embodiments.

In reference to the description of Figure 1, each bank of SDRAM module 16A-C includes a plurality of pages that are accessed by a particular memory access address. Depending on the particular type of request received by arbiter 310 of Figure 4, one or more cycles will be initiated by scheduler 320 of Figure 8 to access the requested page. For example, when scheduler 320 receives a PH request, a request to initiate a read/write cycle is loaded in RWQ 830 of Figure 8. Hence only a read/write cycle will be initiated by scheduler 320 to read from or write to the address location. When scheduler 320 receives a PMb request, a request to initiate both an activation cycle and a read/write cycle will be loaded in AQ 810 and RWQ 830, respectively. Since AQ 810 has a higher priority than RWQ 830, the activation cycle will be initiated before the read/write cycle. When scheduler 320 receives a PMc request, a request to initiate an activation cycle and a read/write cycle will also be loaded in activate queue 810 and read/write queue 830, respectively. After the one cycle turnaround bubble described in the description of Figure 1, scheduler 320 of Figure 8 will initiate an activation cycle followed by a read/write cycle.

If scheduler 320 receives a PC request, a request to initiate a precharge cycle, an activation cycle and a read/write cycle will be loaded in PQ 820, AQ 820 and RWQ 830 respectively. Since PQ 820 has the highest priority, the precharge cycle will execute first, then the activation cycle followed by the read/write cycle. However, since PC requests typically offer the worst latency, a PC request may be converted to an OP request followed by a PMb request. This allows more optimal requests that are received later than the PC to be interleaved between the OP and the PMb.

If scheduler 320 receives a BYP request, conceptually this would be a PMc request. However, it is treated as a PMb request since by definition there are no active chip selects. Scheduler 320 may therefore pass this request through AQ 820 to memory bus 106 in one cycle.

In addition to scheduling and optimizing events on memory bus 106, scheduler 320 may also be configured to control various other SDRAM related activities including: refresh cycles to the SDRAM, initialization and configuration of the SDRAM out of reset and power up and power down of the SDRAM.

Although the system and method of the present invention is described in connection with several embodiments, it is not intended to be limited to the specific forms set forth herein, but on the contrary, it is intended to cover such alternatives, modifications, and equivalents, as can be reasonably included within the spirit and scope of the invention as defined by the appended claims.

INDUSTRIAL APPLICABILITY

This invention is applicable to computer systems.

WHAT IS CLAIMED IS:

1. A bus bridge (102) comprising:

a memory controller (210) for controlling accesses to a memory (104);

a processor bus interface (204) coupled to said memory controller (210), wherein said processor bus interface (204) provides an interface between said memory controller (210) and a processor (101);

a first peripheral bus interface (212) coupled to said memory controller (210), wherein said first peripheral bus interface provides an interface between said memory controller (210) and a first peripheral bus (114);

wherein said memory controller (210) is configured to accept and arbitrate a plurality of incoming requests from said processor bus interface (204) and said first peripheral bus interface (212), wherein said memory controller (210) categorizes said plurality of incoming requests into a page hit request, a page miss bank request and a page miss-different chip select request, wherein said memory controller (210) is configured to provide said page hit request with a higher arbitration priority than said page miss bank request, and wherein said memory controller (210) is configured to provide said page miss bank request with a higher arbitration priority than said page miss-different module select request.

2. The bus bridge (102) as recited in claim 1, wherein said memory (104) is coupled to said bus bridge (102) through a memory bus (106), wherein said memory bus (106) is a synchronous dynamic random access memory bus.

3. The bus bridge (102) as recited in claim 1, wherein said memory controller (210) comprises:

an arbiter (310) which is configured to arbitrate between said incoming requests, wherein said arbiter (310) comprises a plurality of categorizing multiplexers (410) configured to categorize said plurality of incoming requests.

4. The bus bridge (102) as recited in claim 3, wherein each one of said plurality of categorizing multiplexers (410) is a multiplexing unit, wherein each multiplexing unit comprises a plurality of multiplexers (610-650) which are configured to select one request from said plurality of incoming requests.

5. The bus bridge (102) as recited in claim 4, wherein said arbiter (310) further comprises:

a priority select multiplexer unit (420) coupled to receive a plurality of categorized incoming requests from said plurality of categorizing multiplexers, wherein said priority select multiplexer unit (420) is configured to select a priority request to be scheduled based on a fixed priority, wherein said priority select multiplexer unit (420) comprises a plurality of multiplexers (710-790).

6. The bus bridge (102) as recited in claim 5, wherein said memory controller (210) further comprises:

a scheduler (320) coupled to receive said priority request selected by said priority select multiplexer unit (420), wherein said scheduler (320) is configured to schedule said priority request onto said synchronous dynamic random access memory bus, wherein said scheduler comprises a pre-charge queue (820), an activate queue (810), and a read/write queue (830), wherein a precharge cycle is loaded into said pre-charge queue, wherein an activation cycle is loaded into said activate queue, wherein a read/write cycle is loaded into said read/write queue.

7. The bus bridge (102) as recited in claim 1, wherein said memory controller (210) is further configured to categorize said plurality of incoming requests into a page conflict request, wherein said memory controller (210) is configured to provide said miss bank-different chip select request with a higher arbitration priority than said page conflict request.

8. The bus bridge (102) as recited in claim 7, wherein said plurality of incoming requests comprises:

at least one request from said processor bus interface (204);

at least one request from said first peripheral bus interface (212);

at least one request from said second peripheral bus interface (214); and

at least one request from a graphics address remapping table (GART) table walk (330).

9. A method of operating a memory controller (210) comprising:

receiving incoming requests from a processor bus interface (204), and a first peripheral bus interface (212);

categorizing said incoming requests into a plurality of categories, wherein said plurality of categories comprise a page hit request, a page miss bank request, and a page miss-different chip select request;

prioritizing among said incoming requests, wherein said page hit request receives a higher arbitration priority than said page miss bank request, wherein said page miss bank request receives a higher arbitration priority than said page miss-different chip select request.

10. A computer system comprising:

a microprocessor (101) coupled to a processor bus interface (204), wherein said microprocessor (101) generates one or more requests;

a main memory (104) coupled to a memory bus interface;

a bus bridge (102) coupled to provide an interface between said processor bus interface (204), said memory bus interface, a first peripheral bus interface (212), and a second peripheral bus interface (214), wherein said first peripheral bus interface (212) generates at least

one request, wherein said second peripheral bus interface (214) generates at least one request;

wherein said bus bridge (102) comprises a memory controller (210), wherein said memory controller (210) is configured to accept and arbitrate a plurality of incoming requests from said processor bus interface (204), said first peripheral bus interface (212) and said second peripheral bus interface (214), wherein said memory controller (210) categorizes said plurality of incoming requests into a page hit request, a page miss bank request and a page miss-different chip select request, wherein said memory controller (210) is configured to provide said page hit request with a higher arbitration priority than said page miss bank request, wherein said memory controller (210) is configured to provide said page miss bank request with a higher arbitration priority than said page miss-different chip select request.

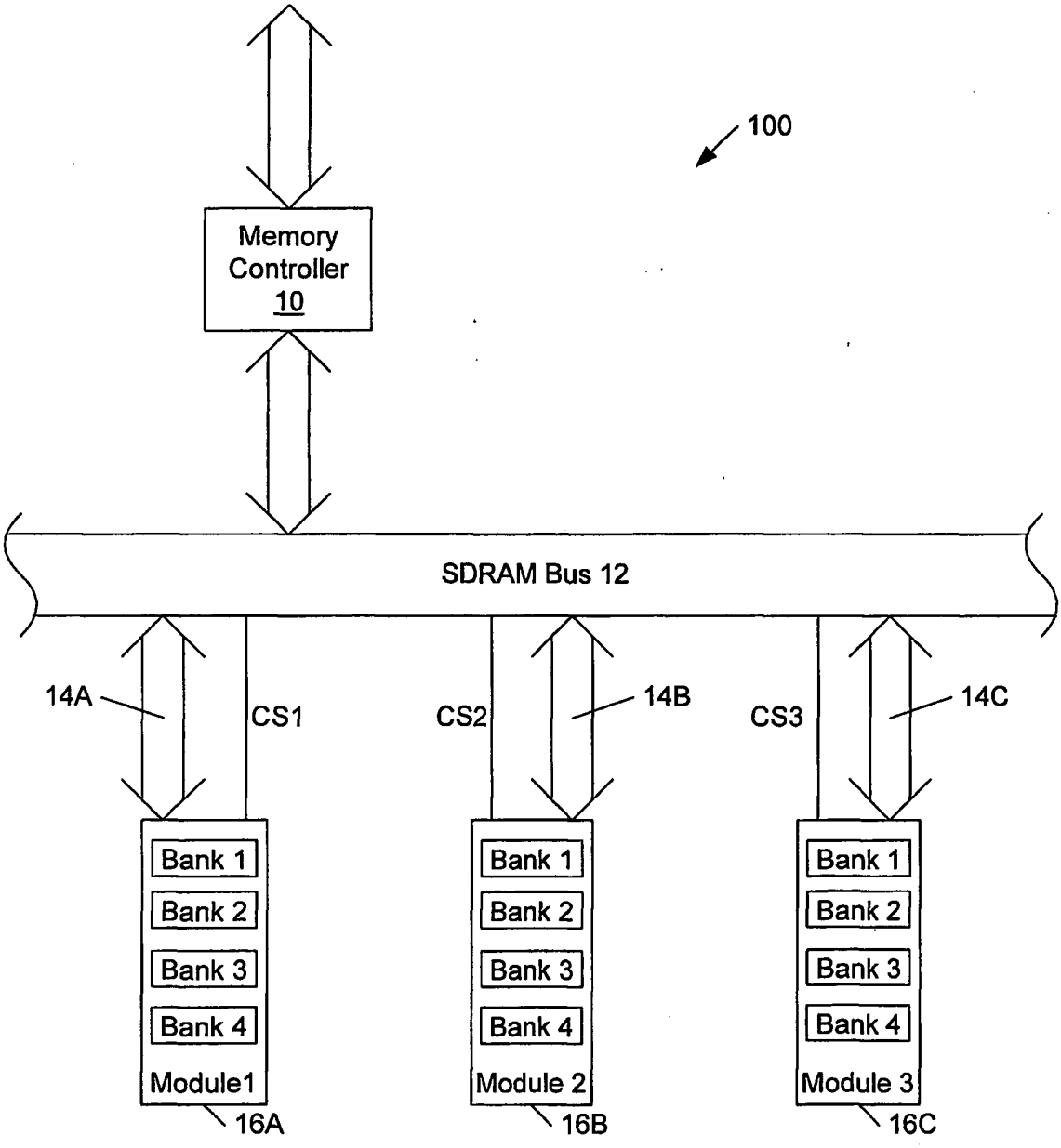


Fig. 1
(Prior Art)

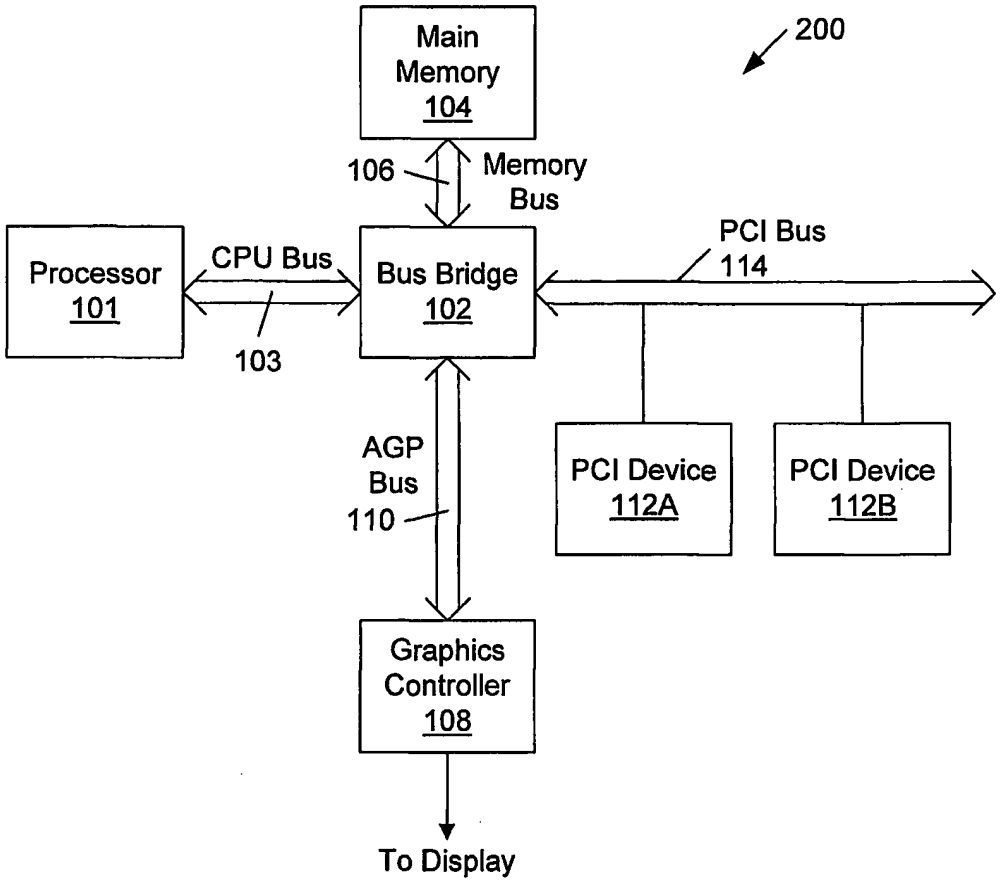

Main
Memory
104
200
106
Memory
Bus
Processor
101
CPU Bus
103
Bus Bridge
102
PCI Bus
114
AGP
Bus
110
PCI Device
112A
PCI Device
112B
Graphics
Controller
108
To Display

Fig. 2

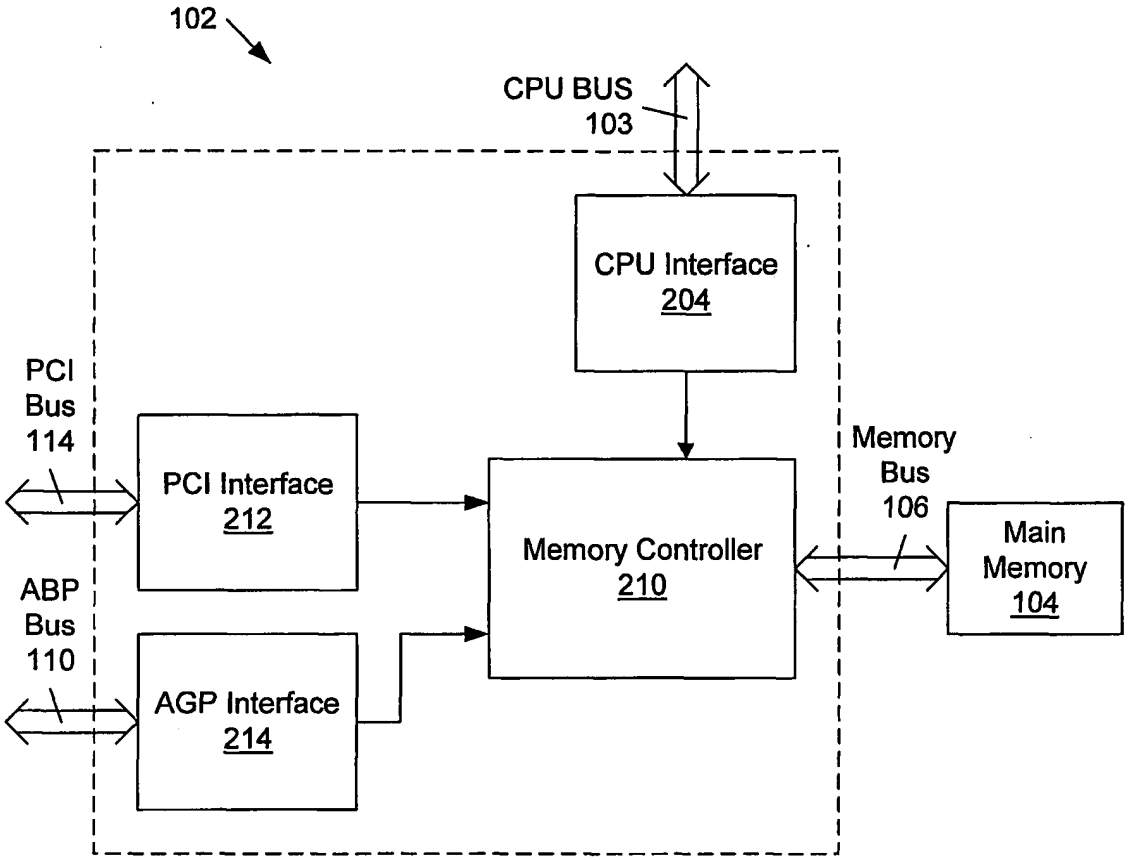

102
CPU BUS
103
CPU Interface
204
PCI
Bus
114
PCI Interface
212
Memory Controller
210
Memory
Bus
106
Main
Memory
104
ABP
Bus
110
AGP Interface
214

Fig. 3

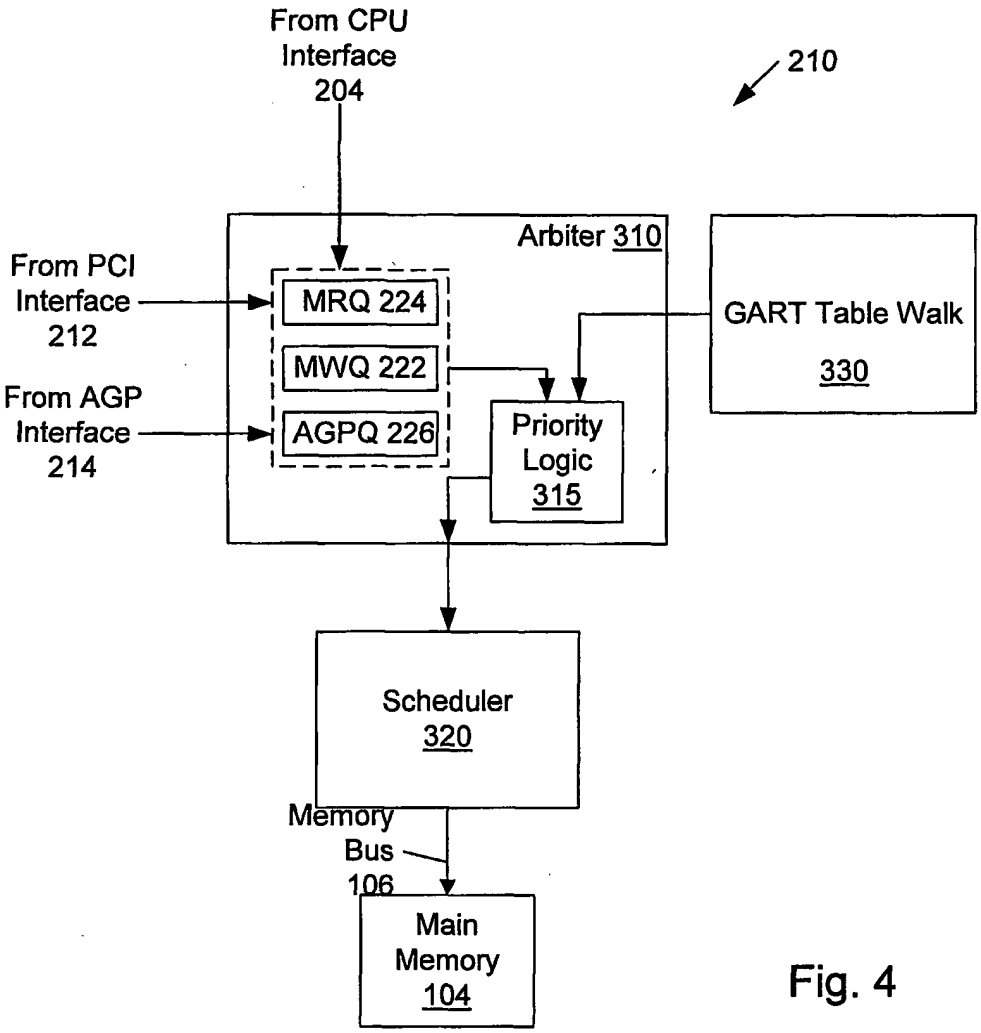

From CPU Interface 204
210
Arbiter 310
From PCI Interface 212
From AGP Interface 214
MRQ 224
MWQ 222
AGPQ 226
Priority Logic 315
GART Table Walk 330
Scheduler 320
Memory Bus 106
Main Memory 104

Fig. 4

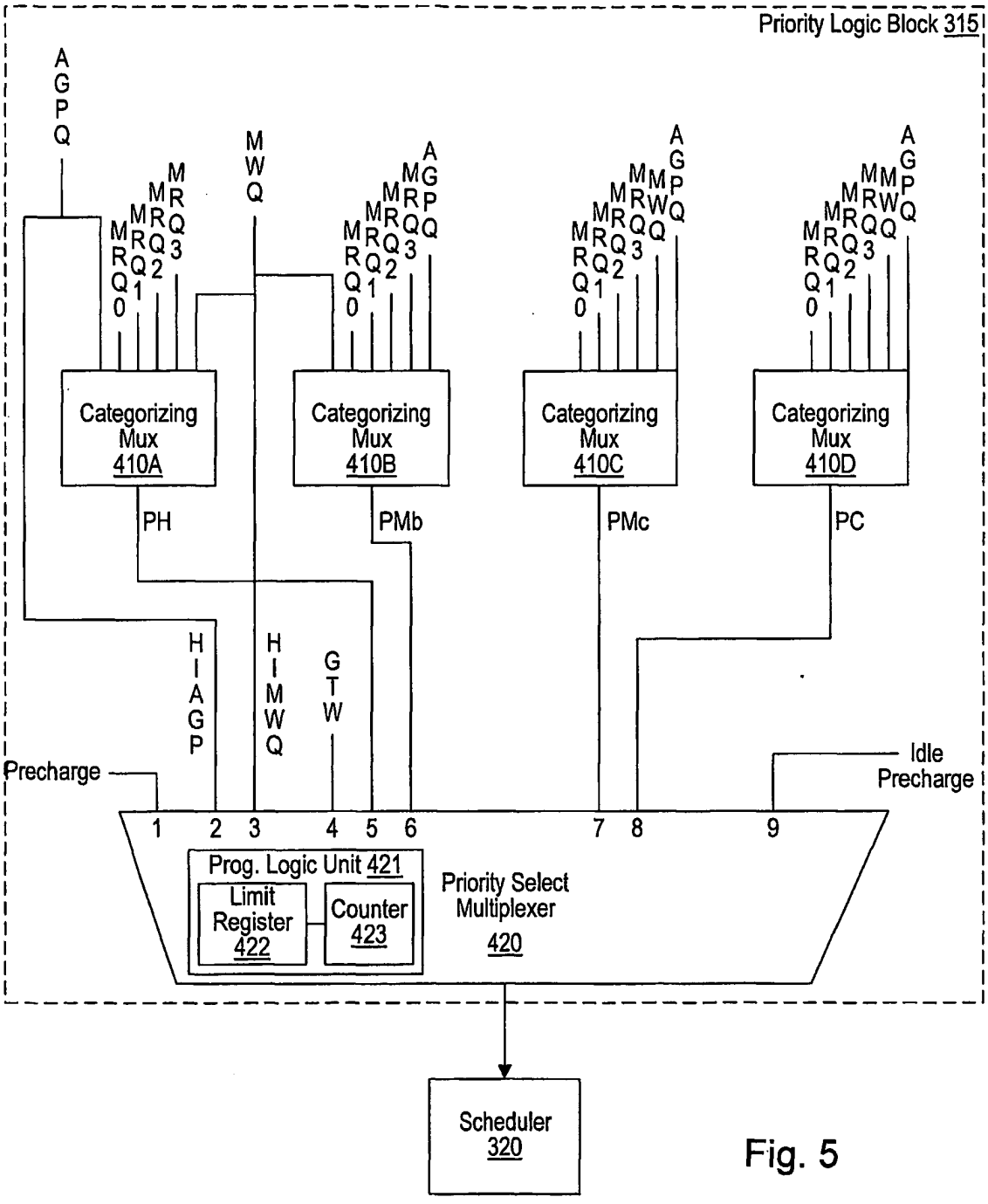

Priority Logic Block 315
AGPQ
MRQ0
MRQ1
MRQ2
MRQ3
MWQ
AMGRPQ
MRWQ
Categorizing Mux 410A
Categorizing Mux 410B
Categorizing Mux 410C
Categorizing Mux 410D
PH
PMb
PMc
PC
HIAGP
HIMWQ
GTW
Precharge
Idle Precharge
1 2 3 4 5 6 7 8 9
Prog. Logic Unit 421
Limit Register 422
Counter 423
Priority Select Multiplexer 420
Scheduler 320

Fig. 5

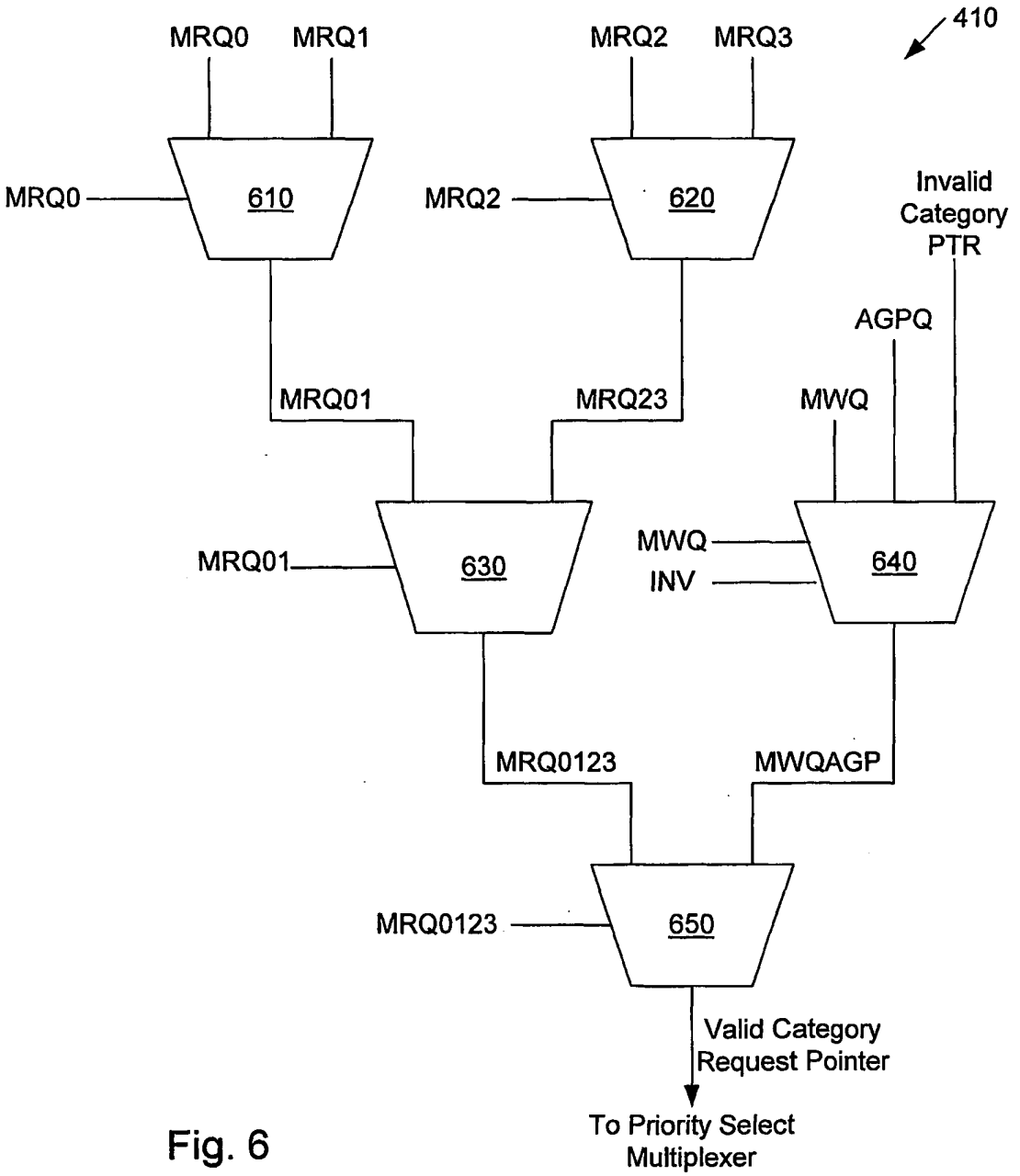

410
MRQ0
MRQ1
MRQ2
MRQ3
MRQ0
610
MRQ2
620
Invalid
Category
PTR
AGPQ
MRQ01
MRQ23
MWQ
MRQ01
630
MWQ
INV
640
MRQ0123
MWQAGP
MRQ0123
650
Valid Category
Request Pointer
To Priority Select
Multiplexer

Fig. 6

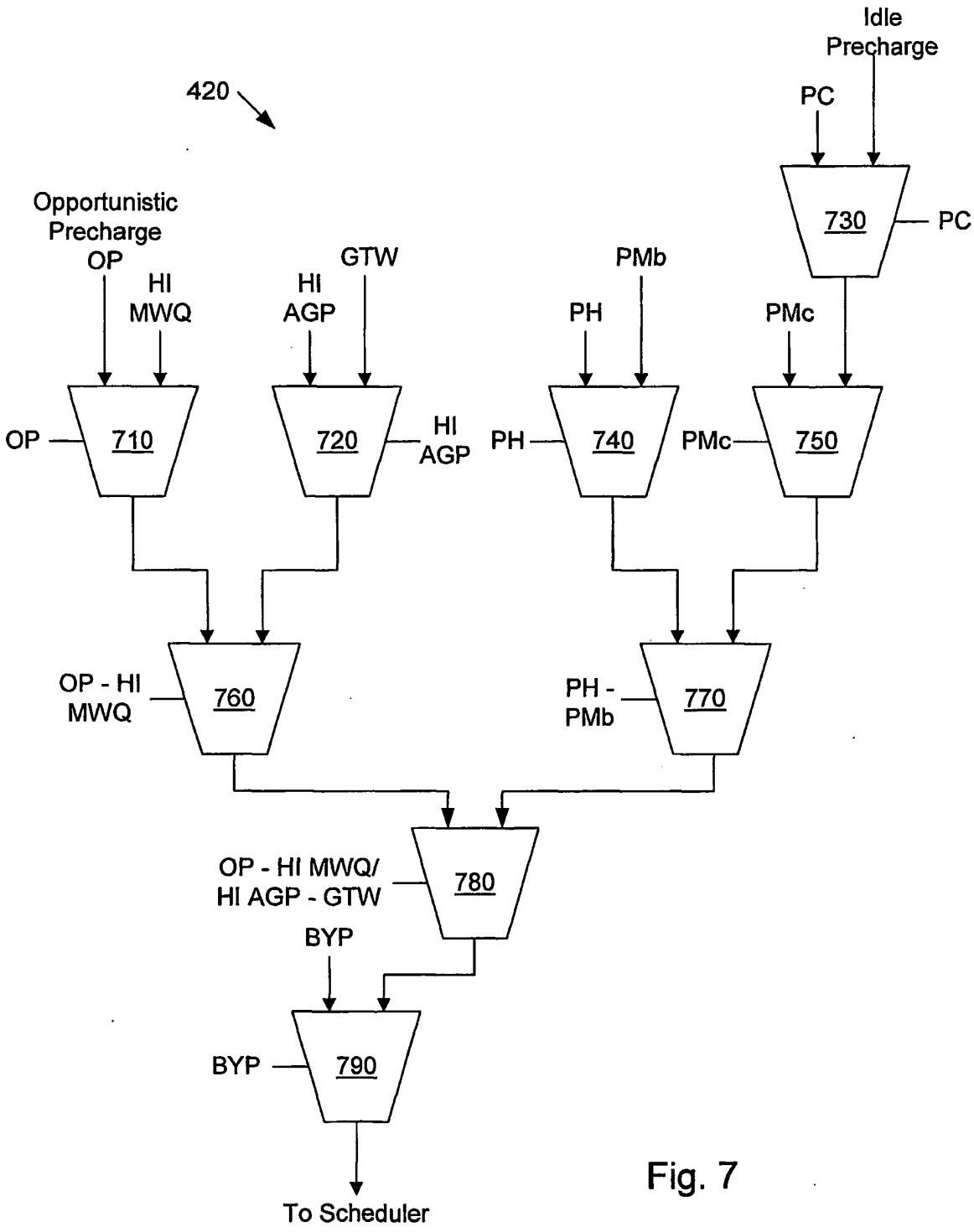

420
Idle Precharge
PC
730
PC
Opportunistic Precharge OP
HI MWQ
HI AGP
GTW
PMb
PH
PMc
OP
710
720
HI AGP
PH
740
PMc
750
OP - HI MWQ
760
PH - PMb
770
OP - HI MWQ/ HI AGP - GTW
780
BYP
BYP
790
To Scheduler

Fig. 7

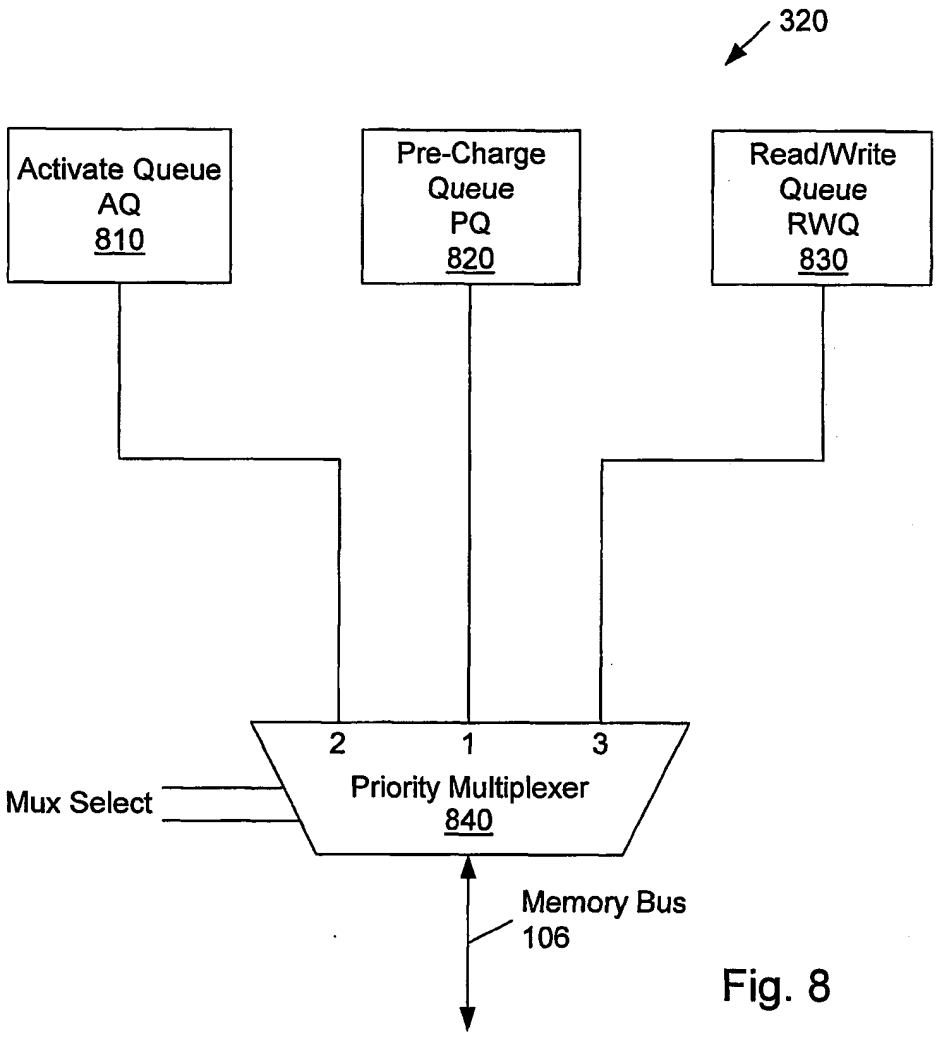

320
Activate Queue
AQ
810
Pre-Charge
Queue
PQ
820
Read/Write
Queue
RWQ
830
2
1
3
Mux Select
Priority Multiplexer
840
Memory Bus
106

Fig. 8

INTERNATIONAL SEARCH REPORT

Int al Application No
PCT/US 00/31963

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F13/18 G06F13/40

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, PAJ, INSPEC, IBM-TDB, COMPENDEX

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	EP 0 924 621 A (COMPAQ COMPUTER CORP) 23 June 1999 (1999-06-23) column 1, line 8 - line 15 column 1, line 29 - line 41 column 5, line 2 - line 8 column 2, line 50 - line 54 claim 1; figure 1	1,2,7-10
Y	US 5 745 913 A (PATTIN JAY C ET AL) 28 April 1998 (1998-04-28) column 1, line 41 -column 2, line 38 claims 1-4; figure 4	1,2,7-10
A	EP 0 391 517 A (DIGITAL EQUIPMENT CORP) 10 October 1990 (1990-10-10) column 2, line 22 - line 48 column 8, line 58 -column 9, line 35; figure 3	1-10

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
18 May 2001	28/05/2001

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Henneman, P

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

In ıal Application No
PCT/US 00/31963

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 638 534 A (MOTE JR L RANDALL) 10 June 1997 (1997-06-10) column 1, line 28 - line 40 column 2, line 34 - line 49 column 3, line 14 -column 3, line 48 column 4, line 35 - line 41 column 17, line 59 -column 18, line 24	1-10

INTERNATIONAL SEARCH REPORT

Information on patent family members

Int. Application No
PCT/US 00/31963

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0924621 A	23-06-1999	JP 11316738 A	16-11-1999
US 5745913 A	28-04-1998	NONE	
EP 0391517 A	10-10-1990	US 5222223 A	22-06-1993
		AT 156278 T	15-08-1997
		AU 628530 B	17-09-1992
		AU 5394790 A	19-12-1991
		CA 1325285 A	14-12-1993
		DE 69031139 D	04-09-1997
		DE 69031139 T	19-02-1998
		JP 2289013 A	29-11-1990
		JP 5026215 B	15-04-1993
US 5638534 A	10-06-1997	AU 693668 B	02-07-1998
		AU 5039896 A	10-10-1996
		EP 0838057 A	29-04-1998
		JP 9055081 A	25-02-1997
		SG 73403 A	20-06-2000
		WO 9630838 A	03-10-1996